IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 17, 2021, accepted August 8, 2021, date of publication August 11, 2021, date of current version August 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104175

Hybrid Constructions of Binary Sequences With Low Autocorrelation Sideobes

MIROSLAV DIMITROV[1], (Graduate Student Member, IEEE), TSONKA BAICHEVA[1], (Member, IEEE), AND NIKOLAY NIKOLOV[2]

[1]Bulgarian Academy of Sciences, Institute of Mathematics and Informatics, 1113 Sofia, Bulgaria
[2]State Agency of National Security, 1000 Sofia, Bulgaria

Corresponding author: Miroslav Dimitrov (mirdim@math.bas.bg)

The work of Miroslav Dimitrov and Tsonka Baicheva were supported in part by the Bulgarian National Science Fund under Contract DH 12/8, 15.12.2017. The work of Nikolay Nikolov was supported in part by the Bulgarian NSF under Contract KP-06-N32/2-2019.

ABSTRACT In this work, a classical problem of the digital sequence design, or more precisely, finding binary sequences with optimal peak sidelobe level (PSL), is revisited. By combining some of our previous works, together with some mathematical insights, few hybrid heuristic algorithms were created. During our experiments, and by using the aforementioned algorithms, we were able to find PSL-optimal binary sequences for all those lengths, which were previously found during exhaustive searches by various papers throughout the literature. Then, by using a general-purpose computer, we further demonstrate the effectiveness of the proposed algorithms by revealing binary sequences with lengths between 106 and 300, the majority of which possess record-breaking PSL values. Then, by using some well-known algebraic constructions, we outline few strategies for finding highly competitive binary sequences, which could be efficiently optimized, in terms of PSL, by the proposed algorithms.

INDEX TERMS Sequences, peak sidelobe level (PSL), digital sequence design, optimization.

I. INTRODUCTION

Digital sequence design plays an important role in various scientific domains, such as radar technology, telecommunications, active sensing systems, navigation, cryptography. One of the desirable characteristic a given binary sequence should possess is a low peak sidelobe level (**PSL**). Some well-known constructions of such sequences includes the Barker codes [1], Rudin-Shapiro sequences [2], [3], m-sequences [4], Gold codes [5], Kasami codes [6], Weil sequences [7], Legendre sequences [8]. Nevertheless, none of the aforementioned constructions guarantees that the generated binary sequence will possess the lowest possible (optimal) PSL value. Thus, currently, initiating an exhaustive search is the only way to reveal an optimal PSL value for binary sequences of some fixed length. Given a binary sequence with length n, the PSL-optimal values for $n \leq 40$ [9], $n \leq 48$ [10], $n = 64$ [11], $n \leq 68$ [12], $n \leq 74$ [13], $n \leq 80$ [14], $n \leq 82$ [15] and $n \leq 84$ [16] are obtained. However, the PSL-optimal values of binary sequences with lengths n greater than 84 are still unknown. This is not surprising, since the search space of the set of all the binary sequences with some fixed length n is 2^n.

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

Since discovering a PSL-optimal value requires significant computational power, from practical point of view, the trade-off between optimality and complexity is justified, i.e. the usage of an algorithm having significantly lower complexity compared to the exhaustive search routine, which is capable of reaching candidates close to the PSL-optimal ones (near-optimal). The state-of-the-art strategies for near-optimal PSL binary sequence construction, as well as ISL (intergrated sidelobe level) sequences construction, include CAN [17], ITROX [18], MWISL-Diag, MM-PSL [19], DPM [20], 1bCAN [21], shotgun hill climbing (SHC) [22] and optimized for long binary sequences hill climbing (HC) [23].

The currently known PSL records for $85 \leq n \leq 105$ are published in [24], and for $106 \leq n \leq 300$ in [22], [25]–[28]. Furthermore, some of the aforementioned works published records for some chosen lengths of $n \geq 300$ as well. For example, in [29] a D-Wave 2 quantum computer was used, altogether with an adiabatic quantum algorithm, for searching binary sequences with low PSL up to lengths of 426.

In Section III, we demonstrate that the hybridization of two distinct PSL-optimizing algorithms could be beneficial to the overall goal of finding near-optimal PSL binary sequences. In fact, during our experiments in Sections IV and V, and by using just a general-purpose processor, we were able to find PSL-optimal binary sequences for all those lengths, which were previously discovered by an exhaustive search only. Then, by using the latest hybrid strategy, record-breaking PSL values for almost all binary sequences with lengths in [106, 300] are revealed. Finally, in Section VI, we investigate the applicability of the proposed algorithm as an extension to some well-known algebraic constructions.

II. PRELIMINARIES

We denote as $B = (b_0, b_1, \cdots, b_{n-1})$ the binary sequence with length $n > 1$, such that $b_i \in \{-1, 1\}, 0 \leq i \leq n-1$. The aperiodic autocorrelation function of B is given by

$$C_u(B) = \sum_{j=0}^{n-u-1} b_j b_{j+u}, \quad for\ u \in \{0, 1, \cdots, n-1\}.$$

We define $C_u(B)$ for $u \in \{1, \cdots, n-1\}$ as a sidelobe level. $C_0(B)$ is called the mainlobe. We define the PSL of B as

$$B_{PSL} = \max_{0<u<n} |C_u(B)|.$$

An m-sequence $M = (x_0, x_1, \cdots, x_{2^m-2})$ of length $2^m - 1$ is defined by:

$$x_i = (-1)^{Tr(\beta\alpha^i)}, \quad \text{for } 0 \leq i < 2^m - 1,$$

where α is a primitive element of the field $\mathbb{F}_{2^m}$, $\beta \in \mathbb{F}_{2^m}$, and Tr is denoting the trace function from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$.

Given an odd prime p, a Legendre sequence L with length p is defined by:

$$L_i = \begin{cases} 1, & \text{if } i \text{ is a quadratic residue mod } p \\ -1, & \text{otherwise.} \end{cases}$$

We denote as $B \leftarrow \rho$ the binary sequence obtained from B, by left-rotating it ρ times. By definition, $B \leftarrow |B| \equiv B$. Furthermore, if b_i is the element of B on position i, we will denote as $b_i^{\leftarrow\rho}$ the element of $B \leftarrow \rho$ on position i.

Let us denote $C_{n-i-1}(B)$ by $\hat{C}_i(B)$. Since this is just a rearrangement of the sidelobes of B, it follows that:

$$B_{PSL} = \max_{0<u<n} |C_u(B)| = \max_{0\leq u<n-1} |\hat{C}_u(B)|.$$

III. PSL PROBLEM REVISITED

Throughout this section, a brief overview of the existing PSL-optimizing algorithms was made. In [23] a comparison of the state-of-the-art algorithms, in terms of algorithm efficiency (the ratio of the beneficial work performed by the algorithm to the total energy invested) and actual effectiveness (the quality of the achieved results) was made. The best results were achieved by **SHC** [22] algorithm, regarding the binary sequences with length less than 300, and **HC** [23], for all the remaining lengths. However, the approximated binary sequence's length, from which HC starts outperforming SHC, is fuzzy and yet to be determined.

In Table 1 a comparison between the most significant components of SHC and HC was made. In summary, both heuristic algorithms are not deterministic, i.e. starting from two identical states rarely results in two identical ending states. The search operator used in both SHC and HC is the single flip operator. Thus, each modification is a simple composition of single flips. One major difference between the two algorithms is their complexity. Indeed, in HC the time complexity of the flip operation is linear, which is a significant advantage compared to the quadratic one to be found in SHC. Another major difference between HC and SHC is the probability of missing (fails to detect) a better binary sequence, which is just 1 flip away from the current position.

As observed in [23], the PSL-optimization process of very long binary sequences is a time-consuming routine, despite the algorithm's linear time and memory complexities. Thus, HC avoids restarts, i.e. re-initializing the starting state with a pseudo-random binary sequence. However, re-initialization appears to be significantly beneficial when dealing with PSL-optimization of binary sequences with relatively small lengths, such as the SHC algorithm.

By considering the observations made above, we have revisited the SHC algorithm:

- The quadratic flip operator was interchanged with the linear flip operator
- The probing strategy (searching for better candidates) was interchanged with the more efficient probing strategy introduced in HC

For convenience, the flip operator from [23] is given in Algorithm 1. The function **Flip** takes three parameters as input:

- f - bit position to be flipped
- Ψ - binary sequence as an array
- Ω_Ψ - the sidelobes of Ψ as an array

The complete pseudo-code of the kernel of the revisited SHC algorithm is summarized in Algorithm 2. For brevity, the following notations were used:

- n - the binary sequence's length
- $\mathbb{T}$ - the threshold value of the instance
- F - a fixed fitness function
- V, V^* - respectively the current best and the overall best fitness value
- c - the counter. The algorithm quits if the counter c reaches the threshold $\mathbb{T}$
- $\mathbb{Z}_n^+$ - the set of all positive integer numbers strictly less than n
- $\mathbb{L}$, $\mathbb{G}$ - binary variables: $\mathbb{L}$ (local) is activated if V is improved, while $\mathbb{G}$ (global) is activated if V^* is improved
- $\mathbb{B}^n$ - the set of all n-dimensional binary sequences with elements from $\{-1, 1\}$
- Q - the quaking function as defined in [23]. For example, if the input triplet of Q is x, L, SL, the function flips x random bits in L, and at the same time, in-memory updating the sidelobe array SL

IV. FITNESS FUNCTIONS

In this section, considering the significant changes made in the SHC algorithm, the fitness function parameters are

Algorithm 1 The In-Memory Flip Introduced in [23]

procedure Flip(f, Ψ, Ω_Ψ)
  $n = |\Psi|$
  $\delta_{min} \leftarrow \min_{(n-f-1,f)}$
  $\delta_{max} \leftarrow \max_{(n-f,f)}$
  if $f \leq \frac{n-1}{2}$ **then**
    for $q \in [0, \delta_{max} - \delta_{min} - 1)$ **do**
      $\Omega_\Psi[\delta_{min} + q] \mathrel{-}= 2\Psi[f]\Psi[n - q - 1]$
    end for
  else
    for $q \in [0, \delta_{max} - \delta_{min})$ **do**
      $\Omega_\Psi[\delta_{min} + q] \mathrel{-}= 2\Psi[f]\Psi[q]$
    end for
  end if
  if $f \leq \frac{n-1}{2}$ **then**
    for $q \in [0, n - \delta_{max})$ **do**
      $\Omega_\Psi[\delta_{max} + q - 1] \mathrel{-}= 2\Psi[f]\,(\Psi[2f - q] + \Psi[q])$
    end for
  else
    for $q \in [0, n - \delta_{max} - 1)$ **do**
      $\Omega_\Psi[\delta_{max} + q] \mathrel{-}=$
      $2\Psi[f]\,(\Psi[\delta_{max} - \delta_{min} + q] + \Psi[n - q - 1])$
    end for
  end if
  $\Psi[f] \mathrel{*}= -1$
end procedure

TABLE 1. A comparison between SHC and HC.

	SHC	HC
Deterministic	No	No
Search Operator	Flip	Flip
Complexity	$O(n^2)$	$O(n)$
Fitness Function	x^4	x^4
Restarts	Yes	No
Missing Probability	> 0	$= 0$

carefully analyzed, re-evaluated, and updated. Given a binary sequence Ψ, both algorithms (SHC and HC) are sharing the same fitness function F, s.t:

$$F(\Psi) = \sum_{x \in \Omega_\Psi} |x|^4 = \sum_{x \in \Omega_\Psi} x^4$$

During our previous experiments, we reached to the conclusion that interchanging the power 4 with larger or smaller value, is respectively too intolerant or too tolerant to the largest elements in Ω_Ψ. However, since significant changes to the kernel of SHC were made, this observation is to be re-evaluated by series of experiments. More precisely, given a fixed threshold $\mathbb{T}$, and the fitness function $\sum_{x \in \Omega_\Psi} |x|^\alpha$, a comparison between the efficiency of different α values is measured.

In Table 2 the results regarding binary sequences with length 100 are given. Each row of the table corresponds to a different experiment. For a more informative measurement of the overall efficiency of the experiments, another variable $V^\triangledown$ was introduced. It measures the median value of all the

Algorithm 2 The SHC Revisited Kernel

procedure SHC(n, $\mathbb{T}$)
  pick $\Psi \in \mathbb{B}^n$
  $V^*, V, \mathbb{G}, \mathbb{L}, c \leftarrow F(\Omega_\Psi)$, 0, True, False, 0
  while $c < \mathbb{T}$ **do**
    $c \mathrel{+}= 1$
    if $\mathbb{G}$ **then**
      pick $r \in \mathbb{Z}_n^+$
      for $i \in [0, n)$ **do**
        flip($(r + i)\,\%n$, Ψ, Ω_Ψ)
        if $V^* > F(\Omega_\Psi)$ **then**
          $V^*, \mathbb{L} \leftarrow F(\Omega_\Psi)$, True
          break
        else
          flip($(r + i)\,\%n$, Ψ, Ω_Ψ)
        end if
      end for
      if $\mathbb{L}$ **then**
        $\mathbb{G}, \mathbb{L} \leftarrow$ True, False
        continue
      else
        $\mathbb{G} \leftarrow$ False
      end if
    else
      pick r $\in \mathbb{Z}_4^+$
      $Q(1 + \mathrm{r}, \Psi, \Omega_\Psi)$
      $\mathbb{G}, \mathbb{L} \leftarrow$ True, False
    end if
  end while
end procedure

best values V^*. More formally, if t_i denotes the thread i of a given experiment $\mathbb{E}$ with $\mathbb{R}$ restarts, and if the best results achieved by t_i is denoted as V_i^*, then

$$V^\triangledown = \frac{\sum_{i \in \mathbb{E}} V_i^*}{\mathbb{R}}$$

At first, the numerical experiments suggest $\alpha = 3$ as a near-optimal value for achieving best results. Indeed, given a binary sequence with length 100, and $(\alpha, \mathbb{R}, \mathbb{T}) = (3, 10^2, 10^4)$, the value of $V^\triangledown$ is smaller compared to the other experiments' values. This observation is more clearly visible throughout the experiments with binary sequences having length 256 summarized in Table 3 and binary sequences with length 500 (see Table 4 and the triplet $(\alpha, \mathbb{R}, \mathbb{T}) = (3, 10^2, 10^4)$ with $V^\triangledown = 11.51$). However, this tendency of $\alpha = 3$ supremacy over integer values of α is not observable throughout larger values of n. As summarized in Table 5, the triplet $(\alpha, \mathbb{R}, \mathbb{T}) = (4, 10^2, 10^4)$ yields better characteristics than $(\alpha, \mathbb{R}, \mathbb{T}) = (3, 10^2, 10^4)$. In fact, the quality of the binary sequences yielded by the triplet $(\alpha, \mathbb{R}, \mathbb{T}) = (4, 10^2, 10^3)$, having $V^\triangledown$ equal to 24.81, is almost the same as those binary sequences generated by the triplet $(\alpha, \mathbb{R}, \mathbb{T}) = (3, 10^2, 10^4)$ with $V^\triangledown = 24.98$. Since the first threshold value (10^3) is ten times smaller than the second one (10^4),

TABLE 2. Efficiency and comparison of various triplets (α, $\mathbb{T}$, 100).

n	α	$\mathbb{R}$	$\mathbb{T}$	V^*	V^∇
100	1	10^2	10^3	7	7.63
100	1	10^2	10^4	6	7.00
100	2	10^2	10^3	6	6.95
100	2	10^2	10^4	6	6.72
100	3	10^2	10^3	6	6.94
100	3	10^2	10^4	6	6.70
100	4	10^2	10^3	7	7.00
100	4	10^2	10^4	6	6.94
100	5	10^2	10^3	7	7.00
100	5	10^2	10^4	6	6.95
100	6	10^2	10^3	7	7.10
100	6	10^2	10^4	7	7.00
100	7	10^2	10^3	7	7.23
100	8	10^2	10^3	8	8.26

TABLE 3. Efficiency and comparison of various triplets (α, $\mathbb{T}$, 256).

n	α	$\mathbb{R}$	$\mathbb{T}$	V^*	V^∇
256	1	10^2	10^3	13	14.66
256	1	10^2	10^4	13	13.94
256	2	10^2	10^3	11	11.98
256	2	10^2	10^4	11	11.72
256	3	10^2	10^3	11	11.92
256	3	10^2	10^4	11	11.51
256	4	10^2	10^3	11	11.99
256	4	10^2	10^4	11	11.84
256	5	10^2	10^3	12	12.22

and given the negligible difference of the binary sequences' quality (0.17), this correlation is particularly beneficial and could be further exploited to reduce the overall time needed for the binary sequences optimization routines.

During the final two experiments, considering the bigger sizes of the binary sequences, the threshold value is fixed to 10^3. However, the data gathered throughout the previous experiments suggested that if we have a triplet (n, $\mathbb{R}$, $\mathbb{T}_1$) measured with V_1^∇, then, given $\mathbb{T}_1 \geq 10^3$ and some threshold value $\mathbb{T}_2 \gg \mathbb{T}_1$, such that the triplet (n, $\mathbb{R}$, $\mathbb{T}_2$) is measured with V_2^∇, then $V_2^\nabla < V_1^\nabla$.

In Tables 6 and 7, triplets of the form (α, 10^2, 10^3) were analyzed, corresponding to binary sequences with respective lengths of 2048 and 4096. It appears that the longer the binary

TABLE 4. Efficiency and comparison of various triplets (α, $\mathbb{T}$, 500).

n	α	$\mathbb{R}$	$\mathbb{T}$	V^*	V^∇
500	1	10^2	10^3	21	23.19
500	1	10^2	10^4	21	22.10
500	2	10^2	10^3	17	17.83
500	2	10^2	10^4	16	17.04
500	3	10^2	10^3	16	16.94
500	3	10^2	10^4	16	16.61
500	4	10^2	10^3	16	17.04
500	4	10^2	10^4	16	16.89

TABLE 5. Efficiency and comparison of various triplets (α, $\mathbb{T}$, 1024).

n	α	$\mathbb{R}$	$\mathbb{T}$	V^*	V^∇
1024	1	10^2	10^3	34	38.50
1024	1	10^2	10^4	34	35.96
1024	2	10^2	10^3	27	28.27
1024	2	10^2	10^4	26	27.12
1024	3	10^2	10^3	24	25.43
1024	3	10^2	10^4	24	24.81
1024	4	10^2	10^3	24	24.98
1024	4	10^2	10^4	24	24.16
1024	5	10^2	10^3	25	25.32
1024	6	10^2	10^3	25	25.98

TABLE 6. Efficiency and comparison of various triplets (α, $\mathbb{T}$, 2048).

n	α	$\mathbb{R}$	$\mathbb{T}$	V^*	V^∇
2048	1	10^2	10^3	58	65.64
2048	2	10^2	10^3	41	44.32
2048	3	10^2	10^3	37	38.27
2048	4	10^2	10^3	36	36.99
2048	5	10^2	10^3	36	36.74
2048	6	10^2	10^3	36	36.91

TABLE 7. Efficiency and comparison of various triplets (α, $\mathbb{T}$, 2048).

n	α	$\mathbb{R}$	$\mathbb{T}$	V^*	V^∇
4096	1	10^2	10^3	99	110.11
4096	2	10^2	10^3	64	68.48
4096	3	10^2	10^3	55	57.47
4096	4	10^2	10^3	53	54.91
4096	5	10^2	10^3	53	54.17
4096	6	10^2	10^3	53	54.16
4096	7	10^2	10^3	53	54.28

sequences is (n), the larger the aggression of the optimization routine should be (α). Indeed, in the case of $n = 2048$, the best value of $V^\nabla = 36.74$ is calculated by using $\alpha = 5$, while in the case of binary sequences with lengths $n = 4096$, the best value of $V^\nabla = 54.16$ is yielded by using $\alpha = 6$.

V. PRACTICAL APPLICATIONS AND RESULTS

The observations made throughout the experiments in Section IV, as well as the more efficient algorithm constructed in Section III, motivated us to revisit the PSL-optimization problem.

A. FINDING PSL-OPTIMAL BINARY SEQUENCES HEURISTICALLY

As previously discussed, binary sequences with lengths up to 84 and PSL-optimal values have been already discovered by using various exhaustive search strategies. This data is particularly beneficial for measuring the efficiency of a given PSL-optimizing algorithm. In other words, given a search space with binary sequences with some fixed length $n \leq 84$, and some PSL-optimizing algorithm $\mathbb{A}$ with a reasonable threshold value, the best results achieved by $\mathbb{A}$ could be compared with the already known optimal PSL values.

During our experiments, we have used a single general-purpose computer with a 6-cored central processing unit architecture, capable of running 12 threads simultaneously. Surprisingly, by using the SHC revisited kernel, as well as a fixed value of $\alpha = 2$, we were able to reach binary sequences with optimal PSL values for each length in [1, 82]. Given the linear time and memory complexities of the algorithm, for the majority of those lengths, the PSL-optimal binary sequences were reached for less than a minute. However, for some border cases, the needed time was few hours. The best results yielded by our experiments are summarized in Table 8. A remark should be made, that we have included just one PSL-optimal binary sequence for a given length. However, for almost each fixed length, the algorithm was able to find more than one binary sequence having an optimal PSL value. The binary sequences are given in a hexadecimal format, by omitting the leading zeroes. In the last column of Table 8, beside the corresponding optimal PSL value of the hexadecimal binary sequence given in column 2, the symbol ⧖ was used to illustrate some approximation of the time needed for Algorithm 2 to reach a PSL-optimal binary sequence:

- ⧖ $\approx$ minute
- ⧖⧖ $\approx$ hour
- ⧖⧖⧖ $\approx$ day

For all other cases, the algorithm was able to reach the optimal PSL for less than a minute, and in some cases, for less than a second.

B. FINDING PSL-NEAR-OPTIMAL BINARY SEQUENCES HEURISTICALLY

In [16] it was shown that there are no binary sequences with lengths 83 or 84 with PSL 4 (or less). For completeness, in Table 9, two binary sequences (with lengths 83 and 84) reached by Algorithm 2 are given. Both possess an optimal PSL value and were reached for less than a minute.

The near-optimal PSL values for binary sequences with lengths from 85 to 105 are found in [24]. However, there is no further information regarding the particular optimization technique that was applied. The authors just stated that *'The searches involved a combination of several global optimization methods'*. Hence, it is difficult to recreate the experiment or, for example, apply the aforementioned mix of unknown optimization techniques to binary sequences with different (greater) lengths. Nevertheless, by using Algorithm 2, we were able to reach the same PSL values for all the binary sequences with lengths from 85 to 105 (see Table 10). It should be mentioned that the binary sequences from Table 10) are different from those that were previously published in the literature.

C. FINDING BINARY SEQUENCES WITH RECORD-BREAKING PSL VALUES HEURISTICALLY

The results achieved throughout the experiments described in Sections V-A and V-B demonstrated the efficiency of

TABLE 8. Reached optimal solutions.

n	Sequence in HEX	PSL
10	37a	3
11	712	1
12	b3	2
13	a60	1
14	2a60	2
15	3dba	2
16	a447	2
17	1c0a6	2
18	2650f	2
19	52447	2
20	87b75	2
21	129107	2
22	14f668	3
23	56ce01	3
24	4a223c	3
25	9b501c	2
26	2e7e935	3
27	24bb9f1	3
28	e702a49	2
29	10e2225b	3
30	2a31240f	3
31	2d079910	3
32	2857d373	3
33	16915cc18	3
34	1a43808dd	3
35	5569e0199	3
36	87885776d	3
37	10c1237a2b	3
38	7caacc212	3
39	29ca6c7c80	3
40	22471e86fa	3
41	7c64d77ade	3
42	4447b874b4	3
43	550e7f99b49	3
44	cb4b8778888	3
45	b6cab731e3f	3
46	16959a2e3003	3
47	69a7e851988	3
48	e6e9bd5bc10f	3
49	103f6eda6ae71	4
50	31dceade9920f	4
51	71c077376adb4	3
52	600dc3cb4cd56	4
53	1671848a940fcb	4
54	2622a797806912	4
55	6006a578ea6933	4
56	61e4b3229420af	4
57	143606103beca35	4
58	215081f5644f2ce	4
59	3b06774134bdf5e	4
60	4df905215263a39	4
61	193c99e12d6010aa	4
62	25695564e679ff83	4
63	707d54b9c99ef690	4
64	d4ef33d372e082be	4
65	1f75f6c8f84c6b50	4
66	28a59401e57b1c993	4
67	5ba4d417723078421	4
68	d155a49d98c7bf7e1	4
69	18ff3eb05d654b6665	4
70	2b5aae6765e79b600f	4
71	8cea0ff5e92cb9726	4
72	dbcf036102615ab2a	4⧖
73	164da9aab5398f1ffe1	4⧖
74	8c9c6dab51e57580f	4⧖
75	5ff692ba8d62f1e3326	4
76	87ad414fa9fcbb99a6c	4
77	fe00861c0d932958aca	4
78	328b457f0461e4ed7b73	4⧖⧖
79	55fae4fdb42732de2ce2	4⧖
80	fe00a22a539352e3659e	4
81	dc9df3ff085a6c3aae53	4⧖⧖
82	2bf0fceee2499527bc61a	4⧖⧖⧖

TABLE 9. Reached optimal solutions - continued.

n	Sequence in HEX	PSL
83	7fc3af0a919735c4b2591	5
84	fa87fce54c5e3d9964a49	5

TABLE 10. Reached near-optimal solutions.

n	Sequence in HEX	PSL
85	1007b4a2f86ae1cc4cb36	5
86	1378ae9166656250f0435f	5
87	1850253c557b83626f3369	5
88	2c43c8691299154d4fbf04	5
89	17b237ec7daea0c1a7d8d4e	5
90	7ca2b17db11f675bf5ad30	5
91	3dffa15d0b98b16c5c65349	5
92	4c9254743cf393b942217f	5⧖
93	1c9cdc87fdfa50e348aab25c	5⧖
94	3c144be5d296b3e65dc46600	5⧖
95	18d0d61707462bd427fedb24	5⧖
96	d02532058d9cf0d019e578aa	5⧖
97	1a542ff2c2ee6feb3b065186c	5⧖⧖
98	1e61c02e1104b6ea5a981cdc9	5⧖⧖
99	71cf7d0426a0646b20d8a972	5⧖⧖
100	191f7308a8fc4fac34a902c90	5⧖⧖
101	1ff41f8ee334912c6cdca8d28a	5⧖⧖
102	40477758e393668697c0fd6ad	5⧖⧖
100	255b559207e991908213c63e3	5⧖⧖
101	1ff41f8ee334912c6cdca8d28a	5⧖⧖
102	40477758e393668697c0fd6ad	5⧖⧖
103	4fbcf31f8fe6d103ea8dacad48	5⧖⧖
104	2e76361a08417ada07987744dd	5⧖⧖⧖
105	199bb906d3e822bc96a4110e1c7	5⧖⧖⧖

Algorithm 2. Thus, we have further launched the algorithm on binary sequences with lengths up to 300. The results are given in Tables 11-16. The binary sequences with record-breaking PSL values are further highlighted with the symbol ▼ (black triangle pointing down). Almost all of the results known in the literature were improved. More precisely, we have improved 179 out of 195 cases. Curiously, for some lengths, we have even revealed binary sequences with record-breaking PSL values, having a distance of two to the previously known PSL record value. We will mark those improvements with a double black triangle symbol. An example of such length is 229.

In [29], the best results achieved by the D-Wave 2 quantum computer for binary sequences with length 128 is PSL 8, while Algorithm 2 could reach PSL 6 (see Table 11). For longer lengths, for example, binary sequences with lengths 256, the best PSL achieved by the D-Wave 2 quantum computer was 12, while during our experiments we reached PSL values of 10. In fact, we reached PSL values of 10 for binary sequences up to 271 (see Table 14). For completeness, since the D-Wave 2 quantum computer is tested on binary sequences with length 426, we have further launched Algorithm 2 on the same length. Surprisingly, the algorithm was able to find binary sequences with PSL values of 17 (the best value achieved by the quantum computer) for less than a second. In fact, it reached PSL values of 16, and even 15, for less than a second as well. However, PSL value of 14 (see Table 17) was noticeable harder to reach (199 seconds). During this optimization routine, and driven by the results provided in Table 4 (since 500 is close to 426), we have updated the α value to 3.

Recently, in [28] a multi-thread evolutionary search algorithm was proposed. By using Algorithm 2 we were able to improve almost all of the best PSL values from the aforementioned paper - usually for less than a second. For example, the best PSL value for binary sequences with length 3000 achieved in [28] is 51. We have launched Algorithm 2 on binary sequences with the exact same length. It should be emphasized (see Tables 6 and 7), that the α parameter should be increased to 6. Record-breaking PSL values of 44 and 43 were reached for respectively 111 and 371 seconds. In Table 17 an example of such binary sequence (2nd row) is given. The last column of the table provides a more quantitative measure of the record: ▼x denotes that the corresponding binary sequences possess a record-breaking PSL equal to $P - x$, where P was the previously known record.

VI. HYBRID APPROACHES FOR PSL-OPTIMIZING PROBLEM FOR LONG BINARY SEQUENCES

The reasoning behind announcing one binary sequence as long, or short, is ambiguous. Measuring the largeness of a given binary sequence is probably more related to the capabilities of the used algorithm than the actual length itself. From a practical point of view, some algorithms, or their implementations, would not even start the optimization (or construction) process, since their computational capabilities (or hardware restrictions) would not be able to process the desired length. For example, as discussed in [29], the usage of a 512-qubit D-Wave 2 quantum computer limits the code length that can be handled, to at most 426, due to a combination of overhead operations and qubits unavailability. Moreover, it was estimated that a 2048-qubit D-Wave computer could handle binary sequences with lengths up to 2000. Hence, the exact fixed value differentiating short from long binary sequences is still unclear.

In Table 18 some detailed time measurements of binary sequences with lengths $2^g - 1$, for $g \in N, g \in [13, 17]$ are given. The binary sequences are specially chosen to exactly match the lengths of the well-known m-sequences, generated by some primitive polynomial of degree g over $GF(2)$ denoted by $\mathbb{M}$ (see [30]) and the binary sequences generated by Algorithm 2 denoted by $\mathbb{A}$. The α parameter was fixed to 4. The last column ($\mathbb{A}$) denotes the time needed for Algorithm 2 to reach the corresponding PSL (s, m, h, and D denote respectively seconds, minutes, hours, and days). Evidently, the longer the m-sequence, the harder for Algorithm 2 to find a binary sequence with a better PSL value is. For example, Algorithm 2 required approximately 3 days to find a binary sequence of length 131071 with lower PSL than the optimal m-sequence having the same size. Given a PSL-optimizing algorithm $\mathscr{A}$ we will reference the length n of a binary sequence as $\mathscr{A}$-long if the expected time from $\mathscr{A}$, starting from a pseudo-randomly generated binary sequence with length n, to reach a binary sequence with PSL p, s.t.

TABLE 11. Binary sequences with near-optimal PSL - part I.

n	Sequence in HEX	PSL
106	35101a2373a0160d982f6b4e39a	6
107	2408504b2beac46b8d93cc85f86	6
108	727184e79679234058155e880bd	6
109	5db00f58363f65c08452544632b	6
110	2b5085f188c82cbb79e1ae25c1bb	6
111	700f7ceb4b8a926c793caafcdcee	6
112	1c62bf5e0e2bf9bdb9db524d921b	6
113	10e8e632f9a52d803cd7eac6eddd5	6
114	3fad9a9fa616431ee6a6b8746ba74	6
115	637c6cdec32bd4cbaecaf2ffe1610	6▼
116	a03feff259d626e9c4f46471a5168	6▼
117	1b33da4cc6d5dc7f8a55c9007cb8f0	6▼
118	23c598f4ac7f6afde47b84c05dd592	6▼
119	60835d6bb25f775d6b588d9e361f81	6▼
120	98cc2e429c2f810668dfdf14bab0b2	6▼
121	178ffe7181c3f443365313724aac95a	6▼
122	30d4e9ae516cf0320ad003177377485	6▼
123	369ec917afe507e53bdc97151138738	6▼
124	f15ce151edfd7f0ca9eb4496d833233	6▼
125	1b8730333bcdf414d92203c581a554a5	6▼
126	3b9275a7ba7661bb8dbf8e078ad41257	6▼
127	2933b32d40937c4b6f08e03a851c2c2a	6▼
128	84528942da6f07e733404ee8ba70c3ae	6▼
129	1f80f99bf3cc5c3d6f1aacd4209aa925b	6▼
130	2678ae07e71929fb587022ed6bfdb576d	6▼
131	3cbf4b091ea86cea277167ac6304c812	6▼
132	410028af0ea52e93f029f908ce74d8c99	6▼
133	10c27978f1888d4fb0a97c9326ecafe97f	6▼
134	3f01b89e464dccaabce38e920492b56810	6▼▼
135	550c944868887c4b7b8709d8263de6c81a	7▼
136	dc789e3aa4f65db16085033ab4b40aee42	7▼
137	1bdfe2817aaa3b39d39daf366d86bc0f49 2	7▼
138	1e618e9ba6c707dc94f05ad723357b2bff d	7▼
139	2bd70f3cde89ad5316439120fe3b9b480b 5	7▼
140	79f8036d08785fbef98ba3b2eb54652eb3 3	7▼
141	fdf5f77808f6cf055b0dd9295c878ad32e 2	7▼
142	343638ce5ed915e8abcc9a0beef8128148 94	7▼
143	554e194c63ca5a65f47de2fd999fc0227e bd	7▼
144	e757f83fd667a6c479d5296908879f6c8d 2e	7▼
145	1051d14d00c893e49498fdba0570862f53 9ca	7▼
146	1c8f3f584efe71220e0da5d4d58d10ed11 ec1	7▼
147	2072a669eade89e6058251c9cc2628a5f5 602	7▼
148	136cbb11363a7078d55f1dc696f217b588 520	7▼
149	152214204e428bf4553661919fe41c0db6 9e18	7▼
150	4fc361d2f104c9510a54c53e9afa612346 7f6	7▼
151	3b93bb695ed592557b82497047438f87c4 318 0	7▼
152	62913b08d6326c46c082e52e3feb0e0b6d 7505	7▼
153	152b80f95df5a4a0e1a2e30cbf68cf76e9 ccdeb	7▼
154	183f0383fe80cccbf38ac495965dae7bd7 9a695	7▼
155	1431f0be9440e92cdd1c4d5680659df937 7adca	7▼
156	bbbd712a19673174fdbc9ad3e78d06f40b b1a84	7▼

TABLE 12. Binary sequences with near-optimal PSL - part II.

n	Sequence in HEX	PSL
157	1444313cfc12546b26e36eb70568dc11b7 06bcf5	7▼
158	18785b52d7074935b31708ef988f769911 040aa7	7▼
159	41c5d5f8d8012c40f00d6ba24d35a539cd 8a573a	7▼
160	7277c5d1140ae6e5638c47ab40937830f2 1b684b	7▼
161	1c9d7e8ec413f2eddacc7be1a45a4318ba 3ab2b46	7▼
162	36280d42653b385e990c70aec3d64845dd 59413e	7▼
163	ff4a2a50fadfb069cb64a79bb8eafdf55e 660cc4	7▼
164	8cbd592237b8e9d5dd7fddb148e13a7c0e f03696c	8
165	bdfe78f96cd0a73e41fa8764667b9e82d1 54d6544	8
166	12c242012b761f803271ab9649f67432ee 288d398a	8
167	349aab4a5752c6459e4cd43f18708fd980 18044fc3	8
168	a18f9ca18bdb8eaf44a84db7f3f92ddd36 0ec23bc4	8
169	f77f9c30338bec86cb76455ec4af4d4394 769e17de	8▼
170	1647c513e17c8b5ac12f169cf4008e77de aeedd71d9	8▼
171	455bce3cd34aa5199a53b3f9900ed684d8 11607828e	8▼
172	8a862f714aa517b5e1d4c9e784b66d0c07 eccfd9f61	8▼
173	9087c81b16785b95b1f63942ab8829d1da e83048267	8▼
174	3a595abedb5fb13e998250683feaa608f1 b10f721e8c	8▼
175	75f9a9db5111b640009d36bc18a71887a8 d4f60f5079	8▼
176	f8b81eb83c80faa526c53d6c43bbb18d34 c2b7df7bb3	8▼
177	ceca7a7c3d3d4ed8893081464daa5d50cf 40905fe630	8▼
178	240603787909825762b567fe0a338e0aeb 85db46e98a4	8▼
179	d27d4a3d8ce26560a137f967fd5b2a22fe 7ea4e9cb1e	8▼
180	45f880bba360f4fe67321649f77be67971 e729d54b4a5	8▼
181	16e90d659806f9ad47ab399481f7975581 4e0cc9805074	8▼
182	2ad575a76ca6eb9f36b207790cec2047db f70dd1f07095	8▼
183	3080a6b5d518e2437cbe03e83276091f3d 9ad5bb717275	8▼
184	bd09e4073d719c5290dc81d3edc090b050 3345a2aaddb7	8▼
185	1b0245da96f15f3faaf6f0e71c5a6c6e2e 7ba4fa2190aff	8▼
186	d6c3079d747a0496d4eab7337a91236c73 cefba0eff4f4	8▼
187	34999dc9c93025871f7aaceb517d0e451c 07b504a75da01	8▼
188	36766a797988a55100a42a91e73c43f005 b76d60705f364	8▼
189	880723487ff2acbc3e65d1eba13327b9a0 5965bd52d14e7	8▼
190	21e50af105ba1d87a44214221d935bba27 35951f776101cf	8▼
191	6122466d46065abb2e2595ed350f45d4a7 f173881f4c33ef	8▼
192	1fbfab7bc285711fb852eb5f00b2ba9c36 98e27cd26a66c9	8▼

TABLE 13. Binary sequences with near-optimal PSL - part III.

n	Sequence in HEX	PSL
193	11e5e2e1ea52cd9c13f6ec031979a99549 b90fb8c2600a288	8▼
194	35d745068d86b74ca0a6d8c73a39676ea7 7bd2b4bc0fc0267	8▼
195	1c841bd699c259b0d801b20e4fd8bebe1c 6567ae3abd08a95	8▼
196	b6a64ce8063c6116f91dd3cfc332f8aac5 f7bdc8a0bad6d2a	9▼
197	2b7ceef5fba16ec29257b30a65a26ac34f 1841ddc7c0e0de7	9▼
198	2da669214cb962a811544e5d3d37a000f8 c0c60dced1ed0bce	9▼
199	1144b275da9c8adb8fffce37c87ba0d2c3 c6bda983f4dc032b	9▼
200	66c30c122f4ee5d8b01ab9155a1ca5afed 0d37d4df0775bd84	9▼
201	82a1c892ca09589a5f1ba194c682ef0f71 d182378a64895ff4	9▼
202	1d045e3d7d3e006c938fb456d5f2a4bf5e 4dcce9c41ca663186	9▼
203	6413fc8964522104171ca948e5d4c4e1cf ade1a82d03b3e640d	9▼
204	6730c61d894ad6db47d7db1707d109a8fd 7e9912cfee2df887d	9▼
205	d24ff6dfd7766450d28c6f1d08aa13c6f5 060b93ef182d5e847	9▼
206	7372ccbe4d517dc500e9ed586a99c9fc60 a442016a06fd0c961	9▼
207	5b92dad3371cc960e08e1993a80ac0a9f5 73c2708165ba02bf5f	9▼
208	d47fffd42e8257a630ef1673359f05eb26 ce173462e0ecb498d2	9▼
209	1bf64d73ea8531230afd6c614fceee5aad 2714cd7c1674125f01e	9▼
210	31da42975a5a3c741f6506fc77598874bf 77e37f2ae29fb1304dd	9▼
211	252e50a7cd40fd82e13aae3096361608b2 3030076fbbd84ca636e	9▼
212	87d63ff093d2c932221b74ae6e9443ac63 33b42e0b890a5754141	9▼
213	54614010e30c87b5366b6baa6400fc7e8b 57067a894e9b3f898e4	9▼
214	e7d4a6d69fda9cf9843db94242a88c5cd7 77cd24165c2f913e0fc	9▼
215	301c7898c56aa56687800ffbf3e65a5787 6867c9426eb3dd5d46d3	9▼
216	d332ccdcab19af1972f93007baf8af8057 c3af4b59e4d040624b52	9▼
217	a87867118f48a6922d161093f015d7f8dd b57c80cb5aedd1b0b177	9▼
218	4c91d36554864c73c5ae223a17dd60ec62 96849685d7fb81f3f881	9▼
219	536a2df324baa32c8488880d9ae152f5bd 0b808ebcf131fc0c293c7	9▼
220	bc39257be78b79101abf2c3edb9b3c01e4 157240d46a6319c5789d2	9▼
221	fadda9f6d109fcc882a91bab8478e6ebc5 713826d19fd06b485a061	9▼
222	1e1a28caa7a070d16e6300965eba9752b3 4e37e66d02139025cfc84f	9▼
223	49e28e14ca6daa3c6fc973368464a08c55 94bf408129cfa607b303d1	9▼
224	b5a435ab97a31e722120bbf812cd251cc7 032281cc0aa29f07f9b66e	9▼
225	1f61bc4168c021782f9e50c6b52a1c546a da0864fd9313b32a9bfae98	9▼
226	3bbbf1c19f6ec551ff0ab982ab334dae01 29a63cc6b58b61e968d0d80	9▼
227	378320814f8021439fe15e5a12add18b76 0cd0788aba8ed3630926333	9▼
228	ac5471683c569456b21141ec539fa32e00 78998f3800d377665b36fa	9▼

TABLE 14. Binary sequences with near-optimal PSL - part IV.

n	Sequence in HEX	PSL
229	a18ea64e0c7d887c6fb51278b686a8b401 66199ef8050906a7c1516b3	9▼▼
230	25474a4ba6e7c3434d1c724ef643ea3181 2c8fa1bbfc877bf6ee5488f7	9▼▼
231	180e1d36289672c3086a1511d58dfeb4a7 f13ca44b44fec5d024664dde	9▼▼
232	886bf85a5bf40b7b2fab51ee8712e2bba7 5b358384435d3ccc99dbf7e6	9▼▼
233	5dda7518a3629e66f6ec3823f6cc6c373b 4bac795efacf416fe1a1ab00	9▼▼
234	f8e141d03a5bb1d91ce20721cdb6207f56 b699d33bf575955694dfa930	10▼
235	3feaa21651ef22c8cb05ab35df33b138a0 8c83e1a1ed24685592035e152	10▼
236	5453ea9ffc1e60c3285de3d07b64a1bcc0 95366d4c437dfccd5d8f4fcea	10▼
237	165567767124fabcb4d08f0da7140abd81 f42e5c9a831dda76fffd894c71	10▼
238	c81ea65bf4b9df2e7f7066454c2d3c8e6a 2841e27963c8229db40a0afd8	10▼
239	6a66b95e25a3cb20e16c7b36b1b22e5988 21242ffc69eeaed03bf9f9d753	10▼
240	dcd3bec7a1856d4ea4febb5c0dcc52e119 ffaa69d4c86df1470530793374	10▼
241	15cabbcb3c965d13d1baf6581833a05593 c6ff73c18dfca9e96272a467f29	10▼
242	2cf51c98f2793804326afb59471b2243a9 12fa50b7abce08ef22607d03941	10▼
243	66346d9c9f2d3393fdeaa0075e7f573ef2 7a1b4b8630b4a322df02f9ba47d	10▼
244	cde4bae1750d2d31e5e3b193df44580f92 245b262ffaaf6e42c6bde7b9532	10▼
245	1af4e7a8ed850811188970ae2af8180736 3afb0113d0f9166b49916df928d6	10▼
246	30b36a460dbd8ab690c173b8d8ca8c0351 cb3a170bba020a9417843dd76dd3	10▼
247	5de6adef6aa7775d3812b0cd7831689b5e 39682e61899e9ba3f039b00e27b4	10▼
248	f41f437cb07cf0a0aadf0c67b3f7fe114c c66b766ccd153185293943089549	10▼
249	10242f665effd3eb4875a1fab42f9d4515 fc9e251dae3c607319a69e49366ef	10▼
250	2007616f89095843f3ced5634bf501cfff 55adb4589658662e8ba374f65c676	10▼
251	275419d5069976e3bdde14b3329284641e 6164276b8012963f4d383e161fcba	10▼
252	1b55a5dadcac2ee8c3ef41026edcc98eab f592878208e314f6349886407e13d	10▼
253	16fa06a49b5776c2a804a3f64b59e4fd20 3a358e8a77d8f79f159d7c34654e60	10▼
254	20d5c99925b7a51f543e49ff428d5d4e54 8a26e1280a1a2d9fc5cc33018c70ce	10▼
255	10008133c4e8b9aa47e1546b8b75a0a4fb cc1d2c7925637235e4866f23d20cf2	10▼
256	6e6053b51d9f80a561e97e2cc13cae1d56 38728f2013377e867fbbee26bada65	10▼
257	1a24b6e6c465cf993425fe01cb10c2ac88 2285c51cea5697d378bc40305c6e753	10▼
258	dafc4a13dbc909c653b76970b24085986f f0fd93e73d6bcd8bba9aae855ce8af	10▼
259	23c7a45f27e3ff8fb66d31f630620d9f6f 959318ea2754cff5256657508bdad2c	10▼
260	94db24992764caf16520a31303c3d0a967 2e74e01e8012d787381aaaeae319def	10▼
261	28d24a7097956e9f7a63b183c0d97211ee 4f99b9f94a3de360ff75f75082fb55d	10▼
262	2547150b862f86ac277033a8d7de1cfd81 8cd1012db7104817cbf15c29924695c9	10▼
263	58333ccc921a4318cbddf299f4a0d055a3 3a13554056c856a9380b4ff0e1c60d3f	10▼
264	7a4dd00f8bbafc5095a2f5f00da7131ba7 d6f7ac4ce20662e388a6b0c21273204b	10▼

TABLE 15. Binary sequences with near-optimal PSL - part V.

n	Sequence in HEX	PSL
265	154ab3ecf9568391efd8918b059f988d67a21805a46107cb6b89bd30f4c47405c51	10▼
266	3c690152ba0daf7d5b4f7a3ee3c88ab33f6bb8252dc786c8ccd668169c4bbc4cfc2	10▼
267	15e1810bfa1308e523b851c7078b2be464f66df69c7492775594b91644a16e77aff	10▼
268	32c38e387faae3e8b74eb7d4675bfa49f500cac6c56b4de44a8b7f9d8372666090b	10▼
269	e6fbaa465ee2a294646b484fbf7d498512837f32dc48de2872f0781741941892680	10▼
270	ca6d5d2e2898349ca6f36814244ad6e2048fc50210d8a0fe07fcd5d7f135261c718	10▼
271	4fd6ffb4ef673619e25b08bdb8157332e615587d1c72ee2d9302c5feb706b0acdab4	10▼
272	eb8a2f2227f60eca7d47a60d44193beef14b2502f3b5a198f69d3ed7dfb4eca72b41	11
273	1b007f99648f37cf3f43ffdb61260d2d33b65231ad1cbc3353a1ec6e4bc5555d5ab95	11
273	1d92f5d3696863c9fa0972f85e9023bf7263e0d0472f3a817d42462388332a9db3bad	11▼
274	2a377a8cd8e836fc187135c97cb4f69fadef3a4367b96014b1b9a79bb40d1120baa75	11▼
275	5991082785400a4fec7053b34aeba361d9542b51c7533d37b28524c29f747f285b8c	11▼
276	af8eb78a4018df61ad9e2d5c980dd38ea4dbd3cc1d37126245796adbfad9dccfe27	11▼
277	4a97467cb36e66d3c4062908017d0aa39c6a04ad0f2f27b6c10b1dbaec226a396dc1d	11▼
278	96655611a994569ea5924430f8fbaace178f1df22f07a48c180bef02336e65223642b	11▼
279	7425c1ec9da091b4d0ee98297cf8a600cbb43c455e0031c4f15c642251892bdbcbd5d7	11▼
280	55bef3e1c6a79c1a03aad609724c2da00bba2ad6484112fe95db18d81f99948c6f0b32	11▼
281	1424e102f4fde1aa05941514283b49ba3e1786dae904facc4c6db8ac6632ef12acc89ce	11▼
282	3e88983536cd657fd8069a3360e796e35adc35cb8ab5c6f0ebeeefee25ad9daecc0681f	11▼
283	5f08ad54acf01756103661f0e35a1c815cd9465bf0909bdcbca2081b5ce79bbb96e74d	11▼
284	d1195f2440f108379fb13357dd894f83ff89e0e313269f6b5f48f675acb1218d5769aeb	11▼
285	1fd2b9a522fa0ba0363cefa32874704a1af558b374eb3eadff9593c7925bbc98e4f662d7	11▼
286	12410ee79818cd60c7230ad510aadec0392e476e0f0a036f167bf2be2cab02c0d6d44d81	11▼
287	63dac2f781e251694b5e9978aa03ca24f2a20ad51bbba930e99d93590608f330099de606	11▼
288	69a4d15a39cd274d62e3f41c235b3b280f0336af0833a646b21eb0a04085c40b5fab1aae	11▼
289	16d9909ffc2421af02de219e1d86e042cdbfaff9be97237531d2e1ab96739b5eab8b166aa	11▼

TABLE 16. Binary sequences with near-optimal PSL - part V.

n	Sequence in HEX	PSL
290	15c7ff0aef22f9dbdf7394c8094b13871ac35a9bcd81a472251e5024efd3605951fa0d157	11▼
291	a1c202c94a731846da997686016197dbcd6a6ca7cc264437646ac0c0fcbd11ff5ae07555	11▼
292	6e7871e089cc8db9274cf3a22f22d3d4523d272db2952ab2ac27188b6fbf47faef01bbdf6	11▼
293	4e6aa8af6a0ead457af0ad0ca55efe940e310e4e6f21b73cf0006124c90360db0b987db6c	11▼
294	1a5b6d16eca315188a6c5271c7a7ab9eb3a5ee0efdaabf07b9578110e7fcfe06ccd0a47ecc	11▼
295	4707bfcc051613d674df4982da568161be90f8cb12bf339535d1a7488f0468c03112ae1157	11▼
296	b71a2ac9b154ef459223e3b03cd2394c7e3c15f1ac6a2272deea2c235a20fb4bdbdf3b7fb4	11▼
297	2ff21574c741f68aa9d0872b97acc757c6fdb374791dee45c4a41c274d6c6df5920062de92	11▼
298	32c7b45b87be1884edfec5712d7e3efcef2e825460a6d5dc1b9d4335581e4e33b454d9126b4	11▼
299	587814353b51d6b1d029f7fe73bd9c88ad984a394357ee609923a3ec1923e0bb4047492ea83	11▼
300	5b2a550cad8a468cac4ac82be6ad333849c865361c6d800818ef9387d6513e8cb81d0f23ff6	11▼

$p \leq \lfloor\sqrt{n}\rfloor$, and by using single general-purpose processor, is more than 1 day. Otherwise, we will reference it as $\mathscr{A}$-short. Throughout the radar literature statements that the asymptotic PSL of m-sequences grows no faster than order $\sqrt{n}$ were frequently made. However, as shown in [31], this assumption was not supported by theory or by data. Nevertheless, it appears that the PSL-optimal m-sequences are very close to $\sqrt{n}$ (see [32]). Thus, the threshold value of $\lfloor\sqrt{n}\rfloor$ is based on the expectation that the optimal PSL value for a given binary sequences with length n is less than $\lceil\sqrt{n}\rceil$.

From now on, we denote Algorithm 2 as $\mathscr{A}$ with fixed α value to 4 if not specified otherwise. During our experiments and by using $\mathscr{A}$, we have reached the conclusion that all binary sequences with lengths n, s.t. $n > 10^5$ are $\mathscr{A}$-long. In this section, we have investigated some hybrid constructions which could be applied in those cases when the binary sequences are $\mathscr{A}$-long.

A. USING $\mathscr{A}$ AS AN M-SEQUENCES EXTENSION

In this section, the following procedure is proposed:

- Choose a primitive polynomial f over F_{2^m}
- Fix an initial element a over F_{2^m}
- Convert f to a linear-feedback shift register $\mathscr{L}$
- Expand the $\mathscr{L}$ to a binary sequence L, $|L| = 2^m - 1$.
- Launch $\mathscr{A}$ with L as an input

The primitive polynomials over F_{2^m} could be calculated in advance. Furthermore, the PSL of L, where L is seeded by

TABLE 17. Example of long binary sequences with significantly better PSL values.

n	Sequence in HEX	PSL	▼
426	3075e0e3e3c1581d2af808dfee48904 226a942d671d897292c4613c5b19a5d d22a6799309414418db4ba724a9fd8f fd1dd109b71493	14	▼3
3000	9c9d1dd018fecf19c744616ad4b166 50e04945bf3f38486f3e52499f8687b d6a090f4b79735ec64f9987f6ac4985 ba941983ecdb9d1d0fd861dfdc7ed4a dd34ac12f08b559aa8c22cf70b4724a d819dbb4ddf4678582db3601786cc56 7f8d290b90cbf46e2939152989bba06 5e1644ec8b1d995e9d8d68221ff5166 66bff43b0a993eaa9ba440f0b79f00e 083acd93b1b64ee5acc52cb1a3bc77e 7c01a14a8a7d8003c62fc5778be6a05 df09b9fc03b70dc2df6850a61ed7045 398c52aa1b5baf036848553d7dd27f8 cb72ed847c6796f7216a975dc497149 ef6eab576508ac77dc3c8837d54d952 1d151694dea17e2bb4969a2c4461616 fafaacb172e35685b3bd63152287a79 e329c65b01a41030bf595ec7ef87188 b37a4d3552e73fadefcdf57b05cc618 904a2fdfd52ff7e8a8c1ea9fdf9db08 957495f01fd6ca7ff219ae3c4624100 d4eee30cc0db5aa8e9f548c31b10593 f138b2c7d22c3f7c16279b7b2f65de7 d17494944967d341c6c0c4e70863b00 201984a	43	▼8

some initial element a over F_{2^m}, could be specially chosen to have the minimum possible value. This is easily achievable by using the following theorems (the proofs could be found in [32]):

Theorem 1: Given a binary sequence $B = b_0 b_1 \cdots b_{n-1}$ with length n, the following property holds:

$$\hat{C}_i(B \leftarrow 1) - \hat{C}_i(B) = b_0\,(b_{i+1} - b_{n-i-1})$$

Theorem 2: Given a binary sequence $B = b_0 b_1 \cdots b_{n-1}$ with length n, the difference $\hat{C}_i(B \leftarrow \rho) - \hat{C}_i(B \leftarrow (\rho - 1))$ is equal to $b_{(\rho-1) \bmod n}(b_{(i+\rho) \bmod n} - b_{(n-i+\rho-2) \bmod n})$.

The aforementioned procedure could be better illustrated by an example. If we fix $m = 17$, we could pick the primitive polynomial $f = x^{17} + x^{14} + x^{12} + x^{10} + x^9 + x + 1$ over $F_{2^{17}}$. Before converting f to a linear-feedback shift register $\mathscr{L}$, we should fix the starting state of $\mathscr{L}$. Throughout this example, a is fixed to the initial state of $\mathscr{L}$:

$$[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1]$$

Then, $\mathscr{L}$ is expanded to L. By using single instruction, multiple data (SIMD) capable device and starting with L, we could efficiently enumerate all 2^{17} different binary sequences generated by all possible starting states, to find the one generating the minimum PSL value. More formally, a value ρ_{max}, s.t. $\forall \rho \;:\; (L \leftarrow \rho_{max})_{PSL} \leq (L \leftarrow \rho)_{PSL}$. Considering f and the fixed value of a, in this specific case the value of ρ_{max} is 15150, or more precisely, $(L \leftarrow \rho_{max})_{PSL} = 363$.

Experiments with initializing $\mathscr{A}$ ($\alpha = 6$) with $L \leftarrow \rho_{max}$, instead of pseudo-randomly generated binary sequences,

TABLE 18. Time required to find better PSL values compared to known results from m-sequences exhaustive search.

g	$n = 2^g - 1$	$\mathbb{M}_n^{\mathbb{F}}$ (PSL)	$\mathbb{A}$ (PSL)	$\mathbb{T}$
13	8191	85	84	19s
13	8191	85	83	23s
13	8191	85	82	28s
13	8191	85	81	1.5m
13	8191	85	80	6.95m
13	8191	85	79	4.37h
13	8191	85	78	8.04h
13	8191	85	77	13.24h
14	16383	125	124	44s
14	16383	125	123	1.16m
14	16383	125	122	4.70m
14	16383	125	121	4.72m
14	16383	125	120	5.30m
14	16383	125	119	14.15m
14	16383	125	118	20.26m
14	16383	125	117	20.37m
14	16383	125	116	1.49h
14	16383	125	115	1.49h
15	32767	175	174	47.27m
15	32767	175	173	47.28m
15	32767	175	172	3.09h
15	32767	175	171	3.10h
16	65535	258	257	9.42m
16	65535	258	256	22.79m
16	65535	258	255	22.80m
16	65535	258	254	22.81m
17	131071	363	362	2.95D
17	131071	363	361	2.95D
17	131071	363	360	2.95D

were made. We were able to repeatedly reach record-breaking binary sequences of length 131071 having PSL equal to 359. The time required was less than 2 minutes, which was a significant improvement over the time required for $\mathscr{A}$ (starting from pseudo-randomly generated sequences) to reach binary sequences with PSL close to 359: approximately 3 days. Leaving $\mathscr{A}$ to work for another 46 minutes it even reached binary sequences of length 131071 with PSL 356.

The proposed procedure, as demonstrated, is highly efficient and is capable to reach binary sequences with $\mathscr{A}$-long lengths and record-breaking PSL values for few minutes. Unfortunately, it is applicable on binary sequences with lengths of the form $2^n - 1$ only. However, throughout the next section, we provide another procedure that is able to generate binary sequences with length p and record-breaking PSL values, where p is a prime number.

B. USING $\mathscr{A}$ AS AN LEGENDRE-SEQUENCES EXTENSION

In this section, the following procedure is proposed:

- Choose a prime number p
- Generate the sequence $L = \left[t_1, t_2, \cdots, t_p\right]$
- For i, s.t. $i \in N$, $1 \leq i \leq p$, and in case i is a quadratic residue mod p, replace t_i with 1. Otherwise, replace t_i with -1.
- Launch $\mathscr{A}$ with L as an input

As the numerical experiments suggested in [32], it is highly unlikely that a Legendre sequence with length p, for $p > 235723$, or any rotation of it, would yield a PSL value less than $\sqrt{p}$. Having this in mind, experiments with initializing

TABLE 19. Time required for $\mathscr{A}$ to reach smaller PSL values, when launched from a rotated Legendre sequence with length 235747 and rotation value 60547.

PSL	T
496	1s
482	6s
462	15s
442	24s
422	9.75m
411	12.4m
410	18.9m
409	23.4m
408	23.7m

$\mathscr{A}$ ($\alpha = 8$) with a rotation of Legendre sequence with length 235747 were made (the next prime number after 235723). Again, by using SIMD-capable devices, we have extracted the PSL-optimal rotation among all possible rotations of a Legendre sequence with length 235747. More precisely, on rotation 60547, a binary sequence with PSL equal to 508 was yielded. Surprisingly, $\mathscr{A}$ was able to significantly optimize this binary sequence. As shown in Table 19, for less than 25 minutes, using only 1 thread of a Xeon-2640 CPU with a base frequency of 2.50 GHz, a binary sequence with PSL equal to 408 was found.

Since $\sqrt{235747} \approx 485.54$, it follows that 408 is significantly smaller than the expected value of 485.54. In fact, for leaving $\mathscr{A}$ for a total of 2.21 hours, a binary sequence with length 235747 and PSL 400, or 108▼, was reached: an example of such binary sequence is provided within the complimentary files.

VII. CONCLUSION

In this work, hybrid strategies for constructing binary sequences with optimal and near-optimal PSL values are suggested. By using the observations made throughout this paper, we were able to reveal binary sequences of almost any length with better PSL values than those known to the literature. As demonstrated, the proposed algorithms are applicable to binary sequences of any length. Our numerical experiments suggest that the optimal PSL of binary sequences with length n is significantly below $\sqrt{n}$.

REFERENCES

[1] R. H. Barker and W. Jackson, *Group Synchronization of Binary Digital Systems in Communication Theory*. New York, NY, USA: Academic, 1953, pp. 273–287.

[2] W. Rudin, "Some theorems on Fourier coefficients," *Proc. Amer. Math. Soc.*, vol. 10, no. 6, pp. 855–859, 1959.

[3] H. S. Shapiro, "Extremal problems for polynomials and power series," Ph.D. dissertation, Dept. Math., Massachusetts Inst. Technol., Cambridge, MA, USA, 1952.

[4] S. W. Golomb and A. Klapper, *Shift Register Sequences*. Laguna Hills, CA, USA: Aegean Park Press, 1967.

[5] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 4, pp. 619–621, Oct. 1967.

[6] T. Kasami, "Weight distribution formula for some class of cyclic codes," Coordinated Sci. Lab., Urbana, IL, USA, Tech. Rep., R-285, 1966.

[7] J. J. Rushanan, "Weil sequences: A family of binary sequences with good correlation properties," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 1648–1652.

[8] A. Pott, *Finite Geometry Character Theory*. Berlin, Germany: Springer-Verlag, 2006.

[9] J. Lindner, "Binary sequences up to length 40 with best possible autocorrelation function," *Electron. Lett.*, vol. 11, no. 21, p. 507, 1975.

[10] J. M. Baden and M. N. Cohen, "Optimal peak sidelobe filters for biphase pulse compression," in *Proc. IEEE Int. Conf. Radar*, May 1990, pp. 249–252.

[11] G. Coxson and J. Russo, "Efficient exhaustive search for optimal-peak-sidelobe binary codes," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 302–308, Jan. 2005.

[12] A. Leukhin and E. Potehin, "Binary sequences with minimum peak sidelobe level up to length 68," 2012, *arXiv:1212.4930*. [Online]. Available: http://arxiv.org/abs/1212.4930

[13] A. N. Leukhin and E. N. Potekhin, "Optimal peak sidelobe level sequences up to length 74," in *Proc. Eur. Radar Conf.*, Oct. 2013, pp. 495–498.

[14] A. N. Leukhin and E. N. Potekhin, "Exhaustive search for optimal minimum peak sidelobe binary sequences up to length 80," in *Proc. Int. Conf. Sequences Their Appl.* Cham, Switzerland: Springer, 2014, pp. 157–169.

[15] A. N. Leukhin and E. N. Potekhin, "A bernasconi model for constructing ground-state spin systems and optimal binary sequences," *J. Phys., Conf. Ser.*, vol. 613, May 2015, Art. no. 012006.

[16] A. N. Leukhin, N. V. Parsaev, V. I. Bezrodnyi, and N. A. Kokovihina, "The exhaustive search for optimum minimum peak sidelobe binary sequences," *Bull. Russian Acad. Sci., Phys.*, vol. 81, no. 5, pp. 575–578, May 2017.

[17] H. He, P. Stoica, and J. Li, "Designing unimodular sequence sets with good correlations-including an application to MIMO radar," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4391–4405, Jun. 2009.

[18] M. Soltanalian and P. Stoica, "Computational design of sequences with good correlation properties," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2180–2193, May 2012.

[19] J. Song, P. Babu, and D. P. Palomar, "Sequence design to minimize the weighted integrated and peak sidelobe levels," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 2051–2064, Apr. 2016.

[20] M. A. Kerahroodi, A. Aubry, A. D. Maio, M. M. Naghsh, and M. Modarres-Hashemi, "A coordinate-descent framework to design low PSL/ISL sequences," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5942–5956, Nov. 2017.

[21] R. Lin, M. Soltanalian, B. Tang, and J. Li, "Efficient design of binary sequences with low autocorrelation sidelobes," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6397–6410, Dec. 2019.

[22] M. Dimitrov, T. Baitcheva, and N. Nikolov, "Efficient generation of low autocorrelation binary sequences," *IEEE Signal Process. Lett.*, vol. 27, pp. 341–345, 2020.

[23] M. Dimitrov, T. Baitcheva, and N. Nikolov, "On the generation of long binary sequences with record-breaking PSL values," *IEEE Signal Process. Lett.*, vol. 27, pp. 1904–1908, 2020.

[24] C. J. Nunn and G. E. Coxson, "Best-known autocorrelation peak sidelobe levels for binary codes of length 71 to 105," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 392–395, Jan. 2008.

[25] A. Dzvonkovskaya and H. Rohling, "Long binary phase codes with good autocorrelation properties," in *Proc. Int. Radar Symp.*, May 2008, pp. 1–4.

[26] K. L. Du, W. H. Wu, and W. H. Mow, "Determination of long binary sequences having low autocorrelation functions," U.S. Patent 8 493 245, Jul. 23, 2013.

[27] W. H. Mow, K.-L. Du, and W. H. Wu, "New evolutionary search for long low autocorrelation binary sequences," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 290–303, Jan. 2015.

[28] G. E. Coxson, J. C. Russo, and A. Luther, "Long low-PSL binary codes by multi-thread evolutionary search," in *Proc. IEEE Int. Radar Conf. (RADAR)*, Apr. 2020, pp. 256–261.

[29] G. E. Coxson, C. R. Hill, and J. C. Russo, "Adiabatic quantum computing for finding low-peak-sidelobe codes," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2014, pp. 1–6.

[30] D. Dmitriev and J. Jedwab, "Bounds on the growth rate of the peak sidelobe level of binary sequences," *Adv. Math. Commun.*, vol. 1, no. 4, p. 461, 2007.

[31] J. Jedwab and K. Yoshida, "The peak sidelobe level of families of binary sequences," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2247–2254, May 2006.

[32] M. Dimitrov, "On the aperiodic autocorrelations of rotated binary sequences," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1427–1430, May 2021.